The Impact of Scaling on a Multimedia Connection Architecture

Eve M. Schooler
USC/Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292
schooler@isi.edu

Abstract

As the last two meetings of the Internet Engineering Task Force (IETF) have shown, the demand for Internet teleconferencing has arrived. Packet audio and video have now been multicast to approximately 170 different hosts in 10 countries, and for upcoming meetings the number of remote participants is likely to be substantially larger. Yet the network infrastructure to support wide scale packet teleconferencing is not in place. These experiments represent a departure from the two- to ten-site telemeetings that are the norm today. They represent an increase in scale of multiple orders of magnitude in several interrelated dimensions.

This paper discusses the impact of scaling on our efforts to define a multimedia teleconferencing architecture. Three scaling dimensions of particular interest include: (*i*) very large numbers of participants per conference, (*ii*) many simultaneous teleconferences, and (*iii*) a widely dispersed user population. Here we present a strawman architecture and describe how conference-specific information is captured, then conveyed among end systems. We provide a comparison of connection models and outline the tradeoffs and requirements that change as we travel along each dimension of scale. In conclusion, we identify five critical needs for a scalable teleconferencing architecture.

Key Words: packet videoconferencing, connection architecture, scalability, multimedia.

1 Overview of a Connection Management Architecture

We have proposed a multimedia connection architecture that has served as the basis for discussion on Remote Conferencing Architectures within the IETF [22]. At the core of the modular architecture is the notion of a *connection manager*, which resides at each end system to coordinate the orchestration, maintenance and interaction of multi-user sessions. Per-site connection managers communicate with peers using a distributed connection control protocol [21]. Conceptually, the connection manager is separate from user interfaces to the system, which sit above it offering services up to the user and relaying requests down from the user. By separating the connection manager from the user interface, conference-oriented tools avoid duplication of effort. This encompasses the management of participation, authentication, and presentation of coordinated user interfaces. The connection manager is also separate from the underlying components, shielding it from the decisions specific to each type of shared media (audio, video, groupware).

The connection manager acts as a conduit for control information not only remotely among peer connection managers, but also among other local conference-related components as depicted in Fig.1. Connection managers are loosely coupled with *media agents* that implement the media processing and data communication functions. With media-specific details relegated to underlying media agents, functional commonality is distilled in the connection manager. The connection manager provides general mechanisms for session-related tasks (connect, invite, etc.) and acts as a broker to share information across media agents (participant lists, admission policies, etc.).

Modularity allows dependencies on particular hardware or communications facilities to be encapsulated within individual components of the system for easier deployment into new environments and offers the connection manager selection among choices in media agent capabilities. Thus, the connection manager's other principal responsibility is *configuration management* of end system heterogeneity. End system differences include asymmetries in available media, codec mismatches, variations in bandwidth capabilities, transport incompatibilities, etc. Accordingly, the connection manager's control protocol negotiates a workable set of capabilities among group members (*e.g.*, quality vs. cost, MPEG vs. H.261).

The intent of the architecture is to facilitate interoperation among users' teleconferencing implementations across the Internet. Therefore, the connection manager is used to capture high-level configuration descriptions from users (*e.g.*, the collection of media in which the user is interested, quality of service preferences, etc.), then conveys the requested configuration to peer connection managers. Each connection manager in turn provides more detailed descriptions to its media agents, which translate the configuration requests into real-time flow specifications for underlying networks [20]. Peer connection managers work to negotiate a suitable configuration, by relying on interactions between each connection manager and its media agents, and between each media agent with underlying network services.

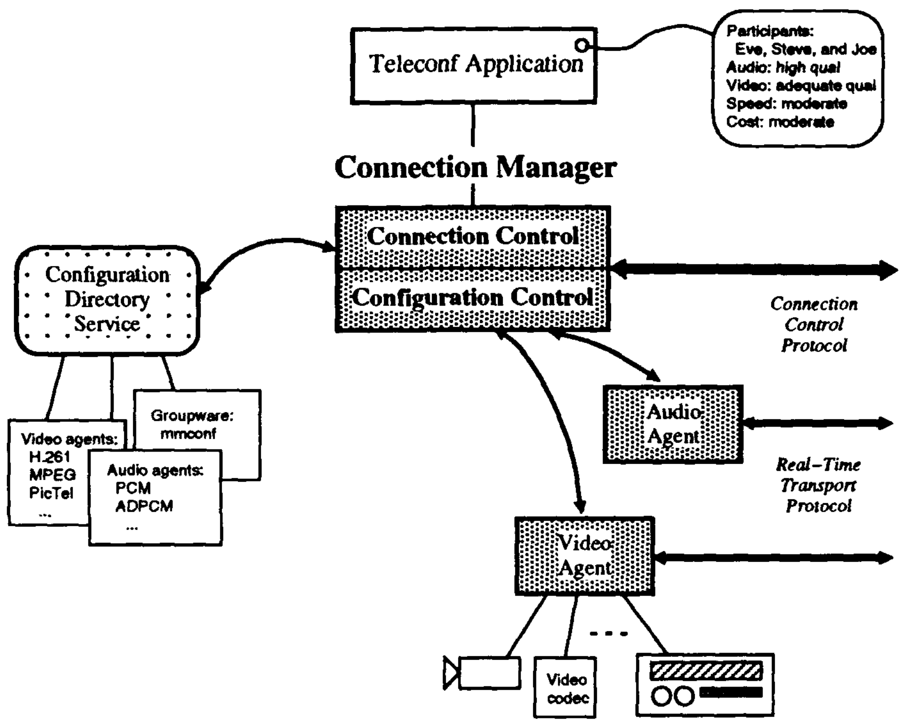

Figure 1. Flow of Control Information

For example, in the simplified scenario in Fig.1, an application asks the connection manager for high quality audio and adequate quality video over moderate speed links for moderate cost. The configuration directory service is consulted by the connection manager and identifies media agents that both meet the specification and are available. In this case, the configuration directory service translates quality, speed and cost into media agents that match encoding/data rate combinations. Once notified, the initiator's local media agents may opt to reserve any devices (cameras, codecs, etc.) and network bandwidth upon which they will rely. The initiator's connection manager then communicates the request to the other participants' connection managers, negotiating over particulars as needed. At this stage, the remote connection managers go through the same process of locating appropriate media agents and reserving the required resources. Finally, each connection manager instructs its local media agents to begin sending data, which means that the media agents establish real-time transport sessions [24]. In a more optimistic scheme, the media agents would wait to reserve resources until all members have actually responded to the initial participation request; delayed reservation however may lead to service denial.

2 The Problem of Scale

Most experimentation with packet teleconferencing systems has been conducted within LAN settings, with few users and with a modest degree of support for simultaneous conferences. In Fig.2, we display a sampling of these systems. The x–axis denotes users per conference, the y–axis the locality of the users (LAN vs WAN), and the z–axis depicts concurrency, or the degree to which each system supports simultaneous teleconferencing sessions.

Although shared workspace applications, such as MMConf, function across WANs, they perform markedly better within LANs [10]. This comes as no surprise since to maintain an actively–changing global view of the workspace, these applications require reliable communication among all users. Typically the application is built on top of an N–by–N collection of TCP/IP streams, which can be problematic within the general Internet. A badly timed network outage or routing problem between one pair of conferees might lead to inconsistency in the shared view. To reconstitute state, a WAN–sensitive session protocol might be layered above the transport to detect and correct peers that are out of synchronization.

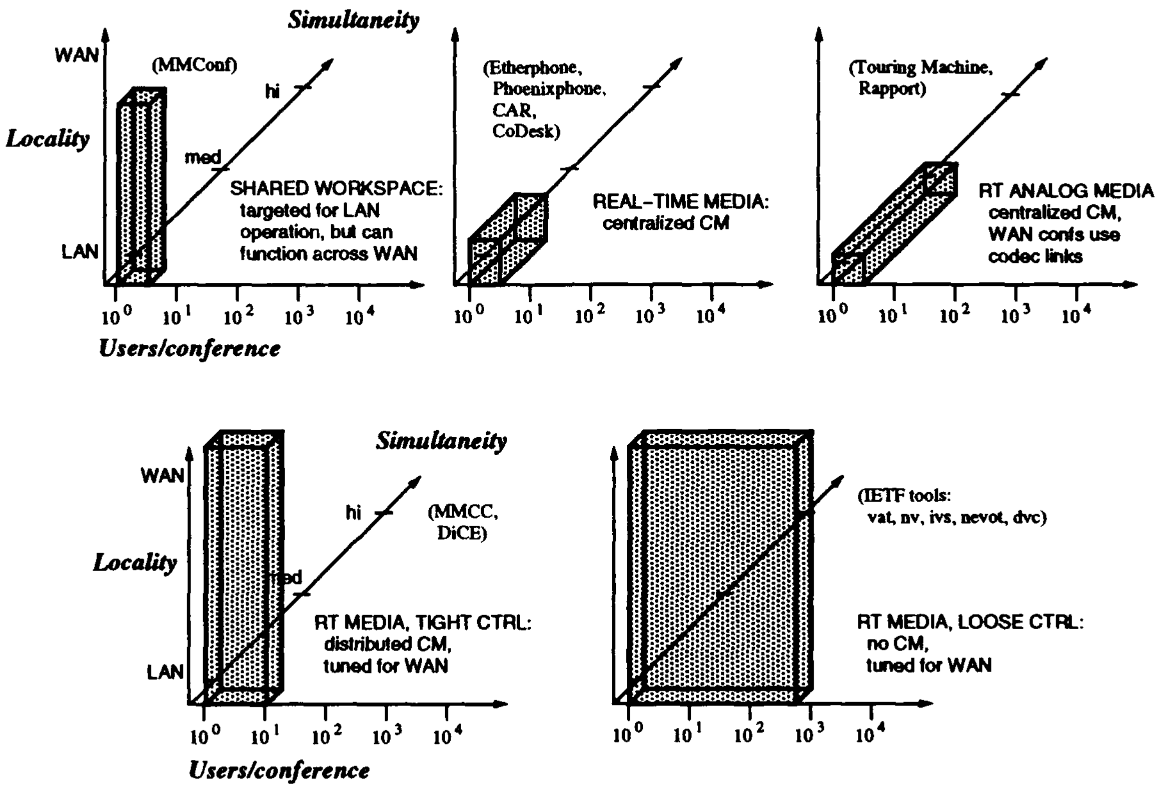

Figure 2. Axes of Scale: Current Teleconferencing Architectures

Real–time teleconferencing systems, such as Etherphone/Phoenixphone, the CAR project and various CoDesk applications, support digital media over a LAN with centralized conference management (CM) [26, 31, 14, 11]. In contrast, the Touring Machine and Rapport represent a class of systems that combine analog media with centralized computer–based session control [2, 1]. In both cases, concurrency is supported, but only as much as the media crossbar switches or the LANs can physically support. To approximate WAN conferencing, analog systems use a proxy to link two distinct LAN communities through a commercial codec.

The second row of diagrams shows systems that are well–equipped for certain aspects of WAN operation by virtue of their decentralized architectures [6, 23, 25, 30]. In addition, MMCC was

designed to accommodate the likelihood in a WAN environment of heterogeneity at the end systems and the need to provide robust sessions across the network [23]. Popular IETF tools, such as LBL's vat, Xerox PARC's nv, INRIA's ivs, UMass' nevot and BBN's dvc, specifically use a lightweight session model to support larger conferences of widely distributed participants [25, 30]. All of these systems, however, are bound in varying degrees by the number of users per conference. None provide explicit support for large numbers of concurrent conferences, due to the Internet's lack of infrastructure for real-time media and wide-scale multicast. These last two classes of systems, formally differentiated by their style of session moderation, will be contrasted in a later section.

As can be seen in all five diagrams, even projects that scale in one dimension, typically have architectural deficiencies in the other dimensions. To understand the problem space better, we analyze how conference requirements change as we travel along each axis of scale.

2.1 Scaling to Large Teleconferences

There is a wide operating range of session sizes and modes. We briefly examine three points along the horizontal-axis that correlate to small, medium, and large conferences. This list is by no means complete but gives a sense of parameters affected by the number of users per conference.

small A small number of participants (ones or a few tens of individuals) allows *impromptu sessions* that are equivalent to our every-day use of the telephone and face-to-face meetings. It allows full connectivity among all users in all media (realtime, non-realtime, control data), flexibility in configuration and negotiation of conferencing parameters, authentication of participants, and the exchange of data encryption keys.

medium As we approach medium sized sessions (hundreds or thousands of participants), we begin to emulate *interactive seminars* that are too large for *N*-way sharing of either data or control. However, impromptu feedback channels are still needed, along with support for dynamic membership. At this size, privacy becomes less practical to provide, even though it might still be desired. The IETF teleconferences were the first medium sized experiments in the Internet [4].

large Large conferences (hundreds of thousands or millions of participants) are analogous to TV *broadcasts*. Information is disseminated in one direction, sessions are pre-arranged or even permanent, and descriptions of sessions remain static. All except the largest conferences should accommodate subconferencing.

2.2 Scaling to a Large, Dispersed User Population

Conferences within LANs often exploit the fixed community of user names, simplified authentication, and homogeneity among end system configurations. It is feasible to maintain a list of user names in a local directory and list them in a calling menu in the user interface. Farther along the axis, the inter-domain problem of obtaining unique user identifiers arises. One naming technique is to combine user names with machine names. A drawback with this approach is that it normally ties the user to a particular location, and with user mobility, the user-to-address mapping requires location independence. However, location-independent addressing will be developed for the use of mobile Internet hosts in a more general context; teleconference user naming will need to build on this capability.

WAN conferencing brings greater likelihood of heterogeneity, less assurance of robustness, increased propagation delay, and movement away from centralized designs to ones that are replicated or hierarchical. Although calling menus might still be useful to list a personal set of aliases, the potentially large community of users makes it impossible to list all possible callees. More likely, inter-domain teleconferencing will rely on a distributed directory to manage the increased naming complexity.

A dispersed and varied user population will come about only when multiple, interoperable teleconferencing systems have been implemented in inexpensive hardware and software. For the

hardware, we depend upon the vendors. For the software, standardized protocols must be developed, and modularity and flexibility in the system architecture must be achieved as primary design goals.

2.3 Scaling to Many Simultaneous Teleconferences

The simultaneity axis is not quite as straightforward, since the raw number of concurrent sessions is uninteresting. More intriguing is that simultaneous sessions generate competition for limited resources, both at end systems and inside the network. From the network's perspective, the main resource under contention is bandwidth, but group addresses, shared multimedia devices and users themselves also become commodities. From the user's perspective, resource discovery is needed to locate these shared commodities, and participation management is needed for call waiting, forwarding, suspension, merging, subconferencing, browsing, and filtering, among other functions.

3 Key Issues: Discussion and Directions

Assessment of the problem space reveals a number of critical needs for a scalable teleconferencing architecture: a range of session control schemes, multicast address management, techniques for bandwidth reduction, a suite of directory services and the detailed codification of heterogeneity. Therefore, we revisit our choice for a session control protocol, discuss the integration of multicast addressing and directory services into our model, and elaborate on additional techniques for bandwidth reduction and heterogeneity.

3.1 A Scalable Session Model and Its Protocols

A variety of researchers have explored frameworks for well-contained conferences [1, 2, 5, 6, 10, 13, 16, 17, 22, 26, 31]; in this *tightly-controlled* model, complete session information is actively shared among and consistently maintained by all conference participants. Participants receive appraisal of who else is involved, acknowledgment that conference state information is current and that communication is reliable. By comparison, IETF multicasts are *loosely-controlled* conferences, where an attendee simply "tunes in" to the agreed-upon multicast address and begins transmitting and/or receiving data. There is no coordination with other end systems, and conference state is constructed asynchronously through the passive (but regular) receipt of control messages from other group members. Even though the IETF experiments used minimal session management, some management functions were simply bypassed by shortcuts. Since there was only one conference, its parameters could be defaulted in the application program, and some functions were handled manually that would need to be automated in a production system.

Because the first scheme relies on full interconnectivity for conference setup and maintenance, it does not scale as well as the latter scheme, which is more lightweight. As we scale up in teleconference size, it is not practical to do the full exchange of status information among all participants for tight control. It would take too long even for one participant to contact all the others, and overload would result if all the participants tried to contact or respond to the same conferee at the same time. An alternative would be to distribute the conference parameters to a set of intermediaries who would each be contacted by a smaller set of participants. A third more extreme alternative would be to post a static set of parameters to a single third party, like a TV guide, or publicly reachable bulletin board.

With many participants, negotiation of parameters also would become impractical because it would take too long to converge upon an agreement and the probability of agreement (finding a common solution) would become small. An alternative is to use a common standard chosen by the conference originator, and only those who can accommodate that standard can participate.

This is not to say that loose-control is the complete solution. For large conferencing, even passive transmission of liveness messages under a loose-control scheme leads to sizable overhead at receivers. Simple communication of participant names on a periodic basis (every 6 seconds) will consume as much bandwidth as a continuous voice channel when the number of participants reaches 300. The period of the updates could be increased or dynamically regulated, but a more explicit control protocol that did not require periodic transmission might be a better solution. The more

apparent disadvantage to loose conferencing is that it lacks support for coordinated group interactions or consensus, *e.g.*, for authentication, floor control, invitations, or quality–of–service negotiations.

These two models represent but two points in a spectrum of services. They roughly are targeted at small and medium–sized conferences. Large conferences require yet a different session model that (in the extreme) has little (to no) setup, maintenance, or communication among participants. Do we devise a session protocol to adapt over the range of conference sizes and modes, or do we create a family of separate protocols for distinct circumstances? The trend for Internet standards is toward simplicity, which might suggest the development of a small number of simple protocols instead of one complex protocol. The characteristics which differentiate these models from one another (level of interconnectivity for session management, flexibility in negotiations, reliability of communication, dynamics of session state, requirements for a consistent global views) need further scrutiny and organization, for they will ultimately influence the outcome. They will dictate the behavior of a more complete session protocol, or define the demarcation points where one protocol ends and another begins.

Thus, in Fig.3 we reframe the architecture in terms of a *scalable session manager* and a *scalable session protocol* upon which it relies. An extension of the original connection manager, the scalable session manager provides a range of conference types beyond and including the tightly–controlled sessions initially supported. We move away from the emphasis on a connection–oriented nomenclature, since some of the session types are connection–less in nature (*i.e.*, stateless), and since we want to avoid confusion between the use of the term "connection" at various levels in the protocol stack.

A final, yet important aspect to the design of a scalable session model is to evaluate options for an underlying transport service that is both reliable and multicast [3, 7, 27]. For conferencing in the large, the transport will need to be lightweight as well. Reliable multicast is needed both in session management and for shared workspaces that are sometimes referred to as groupware. Unlike real–time media that requires a service to support bandwidth guarantees, session control messages and groupware data flows, under many conditions, require a transport service that offers reliability. In Fig.3, we associate different transport needs with the different audio, video and groupware media agents, and imply that the scalable session protocol may be built on top of a transport service similar to that required by a groupware media agent.

3.2 Multicast Address Management

As teleconferences scale up in numbers of users, multicast addressing becomes essential for bandwidth reduction, considering that there is an *NxN* bandwidth explosion for media such as video that normally transmit continuously. As teleconferences scale up along the other axes, management of these group addresses becomes more difficult. For initial IETF experiments, IP multicast addresses have been assigned manually and distributed out–of–band. One complication is that there are a fixed number of multicast addresses. Because most telecollaborations will be transient, address assignment and re–assignment will be highly dynamic. A global scheme is required to avoid unwanted address collisions and to promote reasonable address space sharing. A plan is presented in [24] to partition addresses among a hierarchy of multicast address servers; addresses are borrowed from other servers of greater than or equal stature in the hierarchy, and servers re–use addresses by exploiting locality. To offload dynamic addressing mechanisms, we can make use of fixed multicast addresses for static conferences and use unicast addressing in point–to–point calls.

We envision a local multicast address server being responsible for a single LAN. As shown in Fig.3, the request for a multicast address comes from an individual media agent, or comes from the session manager if the address is being used to send control messages or to multiplex more than one media type. For conferences that are not publicly registered, the session manager distributes the multicast address(es) as part of the session configuration process.

3.3 Techniques for Bandwidth Reduction

Conferencing in the large requires network resource management mechanisms to avoid congestion. Those mechanisms will have to scale to track many connections or flows at once, perhaps using some form of aggregation. Other research projects are working on these problems, and we expect to test and to integrate their solutions as they become available [8, 9, 12, 28, 33]. Specifically, the session manager will collect conference operating parameters from the user interface and will deliver them to these lower-level mechanisms for translation into flow specifications [20]. Before these mechanisms are deployed, it will be possible to use lightly-loaded networks as they are.

While multicasting reduces bandwidth usage by senders, in N-way conferencing the receiver is still faced with a bandwidth N times that of the sender. Thus, mechanisms are also needed for reductions at receivers. A receiver may only want to process M of N streams it is sent, or may have a problem decoding and presenting all the information (*e.g.*, video windows, text aliases for conferees).

One solution is to allow only some fraction of the sources to transmit at any one time. Other researchers have suggested a market-based approach wherein a source is enabled to transmit only if there is a sufficient number of receivers requesting that source [15]. A limitation of the market-driven scheme is that the data from an enabled source would still go to all receivers, including those that had not requested that source. A more general solution is to allow the decisions about what traffic goes where to be made hierarchically, not just at the sender. That is, there may be enough bandwidth (and demand) for one source's traffic to go to some destinations, but not to others. It would be possible to set up separate multicast trees from each source to exactly the set of receivers desiring that source. However, for a large teleconference, that might require too many network resources (such as IP multicast addresses).

We propose *application-level combination nodes* that work in conjunction with participant sources and sinks -- to avoid wasting network bandwidth by deferring reduction decisions until data arrives at the receiver. They would act to hierarchically combine media streams at the application level as they head toward the receivers. These include software or hardware modules that embed functions for: mixing, as with audio streams; compositing, assembling the interesting pieces of several video flows into a single flow; selection, by a sender (chairperson) or receiver (individually tailored); translation, between encodings; reduction, when scalable coding is used; and combinations of these operations along the path from senders to receiver. Multiple combination operations might occur at different points along the path to incorporate additional sources, and the combinations may change over time based on control inputs from the receivers.

Combination nodes are likely to be separate from the end systems involved in the conference. As such, they must be incorporated into all aspects of session management, addressing and routing. They are likely to be described in terms of the function(s) they perform, to act as shared resources in the network and be located at branching points in the spanning trees of multicast routes. The drawbacks of using a combination node are control/routing complexity and increased transmission delay. Fortunately, necessity sometimes decides for us, as in the case of a slow link. A mixer upstream from the slow link would be located, then used to combine several streams into one to circumvent bandwidth limitations that would otherwise prohibit or restrict conference participation. The system behaves similarly when there are incompatibilities between end systems due to heterogeneity. For this case, a translator might be used to go between coding formats.

In either event, the component that integrates combination nodes into the architecture is the *resource synthesizer*. It is intended to sit between the scalable session manager and the configuration directory service (see Fig.3). The quality of the service it provides is somewhat dependent on the configuration directory service, like the other directory services, belonging to a larger hierarchy of information that extends beyond the local domain. The presence of a resource synthesis hierarchy raises questions about who owns combination nodes and who pays for them. A controversial question to resolve will be, under which set of circumstances is it more appropriate to perform these combination functions at the application level or at the network level?

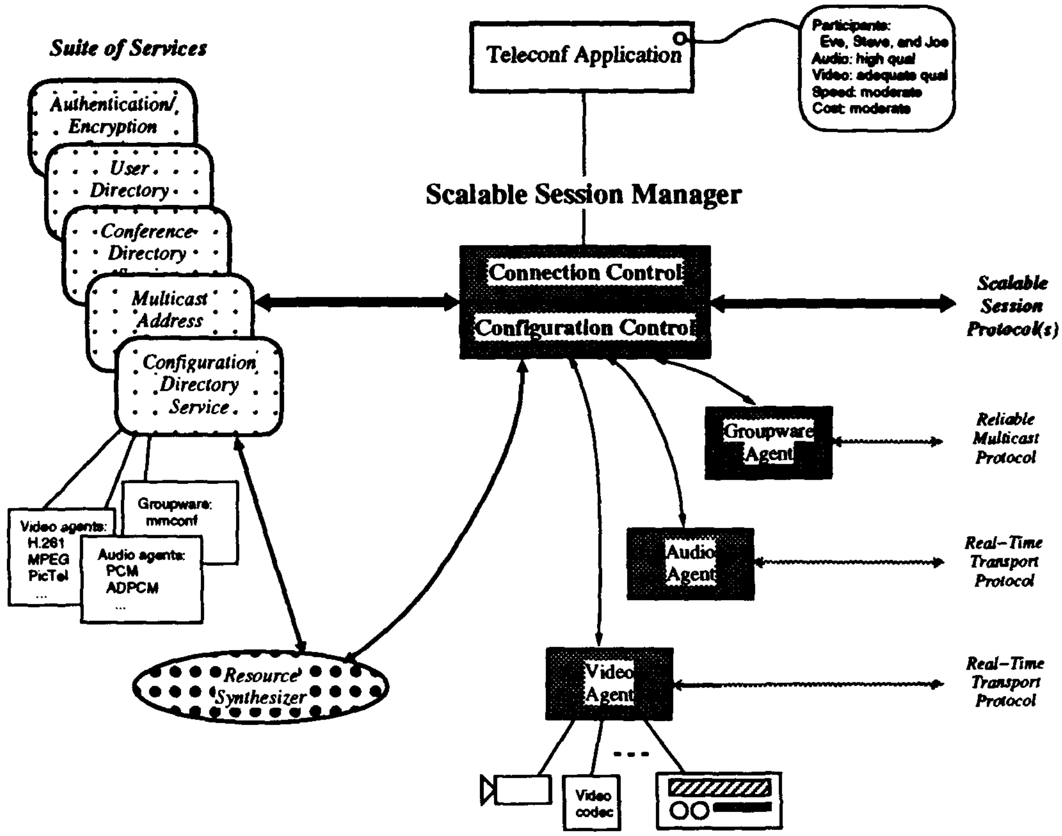

Figure 3. Architecture Components for Scalable Teleconferencing

3.4 A Suite of Directory Services

As depicted in Fig.3, a scalable teleconferencing architecture relies on a whole suite of services, some of which are directory services. Resource discovery is needed to locate inter-domain (and potentially mobile) users, to access conference names and parameters for large private and public teleconferences, and to keep dynamic information on the descriptions and availability of combination node functionality. There is no need to build these directory capabilities from scratch. Rather, the applicability of Prospero, X.500, and/or DNS for maintaining and distributing various attributes of Internet teleconferencing will need to be explored [18, 19, 32]; such a service must support highly dynamic information, replication, and privacy enhancements.

3.5 Codification of Heterogeneity

A *configuration language* for Internet teleconferencing must support highly detailed configuration descriptions, if a connection manager is to provide an abstraction beneath which we truly hide the details of end-system heterogeneity. Although we know that configuration translations occur en route from users to flow specification at local and remote end systems, we concede that much more work needs to be done to define useful configuration descriptions at each stage of the process.

Thus far, the idea of a configuration language has been applied only to negotiations among participants in the event of end system heterogeneity [23]. As the language is in the beginning stages of development, negotiations are still quite rudimentary and are based entirely in terms of a <media, encoding format, data rate> tuple. With exposure to a larger community of users and

domains, we expect to discover a fuller spectrum of configuration parameters that will need representation in the configuration language.

For instance, codification of heterogeneity is needed to support resource synthesis. Of utmost importance are extensions to support combination node descriptions. This is likely to lead to communication classifications (e.g., *1*–to–*N*, *N*–to–*N*, *1*–to–*4*), which we believe will be beneficial to describe additional conference services and modes. These classifications would also provide a basis for the development of a less implementation–dependent conference setup and configuration language, focusing on the operations of the multiparty connection, rather than the particulars of parameterization [29].

There is also the need for configuration descriptions for quality of service at different levels of abstraction. Although a user might make quality of service choices from knobs in the graphical user interface (with markings such as high resolution video or CD quality audio), these selections need translation into media agents capabilities, which in turn require a mapping into network–level flow specifications. The configuration language should support different degrees of expressiveness.

4 Summary

Thus far, few teleconferencing systems address issues of scale. Experiments such as the IETF audiocasts and videocasts are some of the first large scale packet conferences, and these have exposed a number of unsolved problems. We have identified several key architectural components and features that are missing from these experiments, and that would be needed for a more complete solution. We propose extensions to our earlier teleconferencing architecture and protocols to integrate these components.

We predict that over the next ten years, personal teleconferencing is going to become a major source of network traffic. We anticipate that its ubiquity will entitle it to be coined the "Email of the 90s". In the past, multimedia conferencing has served as a driver for Internet technology. Its continued role as such and its viability as a widespread vehicle for telecollaboration depend on how well it can scale.

5 Acknowledgments

I am grateful to the reviewers for their comments. Many of the ideas described in this paper stem from conversations with Steve Casner and other colleagues at ISI, discussions held within the IETF Audio/Video Transport working group, the RemConf Architecture and Conference Control BOFs, deployment and evaluation of prototype implementations [23], and a collection of comments on [21, 22]. IETF experimentation will continue to drive issues on scalability and will motivate more complete architecture and protocol solutions.

This research was sponsored by DARPA under Fort Huachuca contract number DABT63–91–C–0001. The views and conclusions in this documents are those of the author and should not be interpreted as representing the official polices, either expressed or implied, of the Defense Advanced Research Projects Agency of the U.S. Government.

An earlier version of this paper appeared in the *Proceedings of the Third International Workshop on Network and Operating System Support for Digital Audio and Video*, San Diego, CA (Nov 1992).

6 Bibliography

[1] Ahuja, S.R., Ensor, J., Horn, D., "The Rapport Multimedia Conferencing System", *Proceedings of the Conference on Office Information Systems*, Palo Alto, CA (Mar 1988).

[2] Arango, M., et al, "Touring Machine: A Software Platform for Distributed Multimedia Applications", *Proceedings 1992 IFIP International Conference on Upper Layer Protocols, Architectures and Applications*, Vancouver, Canada (May 1992).

[3] Birman, K., Schiper, A., Stephenson, P. "Lightweight Causal and Atomic Group Multicast", Technical Report TR90–1105, Department of Computer Science, Cornell University (1990).

[4] Casner, S., Deering, S., "First IETF Internet Audiocast", *ACM SIGCOMM Computer Communications Review*, Vol.22, No.3 (July 1992).

[5] Chang, Y., Whaley, J., "Remote Conferencing Architecture", *Proceedings of the 24th Internet Engineering Task Force*, Teleconferencing Architecture BOF, Boston, MA (July 1992).

[6] Chen, M., Barzilai, T., Vin, H.M., "Software Architecture of DiCE: A Distributed Collaboration Environment", *Proceedings 4th IEEE ComSoc International Workshop on Multimedia Communications*, Monterey, CA (Apr 1992).

[7] Cheriton, D., "VMTP as the Transport Layer for High-Performance Distributed Systems", *IEEE Communications Magazine* (June 1989).

[8] Clark, D., Shenker, S., Zhang, L., "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism", *Proceedings ACM SIGCOMM '92*, Baltimore, MD (Aug 1992).

[9] Crowcroft, J., Ballardie, Tsuchiya, P., "Inter-Domain Multicast Routing", Working Draft (Oct 1990).

[10] Crowley, T., Milazzo, P., Baker, E., Forsdick, H., Tomlinson, R., "MMConf: An Infrastructure for Building Shared Multimedia Applications", *Proceedings of the Conference on Computer Supported Cooperative Work*, CSCW '90, Los Angeles, CA (Oct 1990).

[11] Eriksson, H., presentation in A/V Transport WG, 24th Internet Engineering Task Force (July 1992).

[12] Ferrari, D., Banerjea, A., and Zhang, H., "Network Support for Multimedia: A Discussion of the Tenet Approach", Technical Report TR-92-072, Computer Science Division, University of California at Berkeley (Nov 1992).

[13] Garcia-Luna-Aceves, J.J., Craighill, E.J., Lang, R., "An Open-Systems Model for Computer-Supported Collaboration", *Proceedings 2nd IEEE Conference on Workstations* (Mar 1988).

[14] Handley, M.J., "The CAR System: Multimedia in Support of Collaborative Design", Technical report, Department of Computer Science, University College London, UK (Oct 1992).

[15] Jacobson, V., Lawrence Berkeley Laboratory, electronic mail (Dec 1991).

[16] Lauwers, J.C., Lantz, K.A., "Collaboration Awareness in Support of Collaboration Transparency: Requirements for the Next Generation of Shared Window Systems", *Proceedings CHI '90 Conference on Human Factors in Computer Systems* (1990).

[17] Leung, W.H., Baumgartner, T.J., Hwang, Y.H., Morgan, M.J., Tu, S.C., "A Software Architecture for Workstations Supporting Multimedia Conferencing in Packet Switching Networks", *IEEE Journal on Selected Areas in Communications*, Vol.8, No.3 (Apr 1990).

[18] Mockapetris, P.V., "DNS Encoding of Network Names and Other Types", RFC 1035 (Apr 1989).

[19] Neuman, B.C., "The Prospero File System: A Global File System based on the Virtual System Model", *Computing Systems*, Vol.5, No.4 (Fall 1992).

[20] Partridge, C., "A Proposed Flow Specification", RFC 1363 (Sept 1992).

[21] Schooler, E.M., "The Connection Control Protocol: Specification (v1.0)", Technical Report, USC/ISI, Marina del Rey, CA (Jan 1992).

[22] Schooler, E.M., "An Architecture for Multimedia Connection Management", *Proceedings 4th IEEE ComSoc International Workshop on Multimedia Communications*, Monterey, CA (Apr 1992).

[23] Schooler, E.M., "Case Study: Multimedia Conference Control in a Packet-switched Teleconferencing System", to appear in *Journal of Internetworking: Research and Experience*.

[24] Schulzrinne, H., "A Transport Protocol for Audio and Video Conferences and other Multiparticipant Real-Time Applications", Working Draft (Oct 1992).

[25] Schulzrinne, H., "Voice Communication Across the Internet: A Network Voice Terminal", Department of Electrical and Computer Engineering, Department of Computer Science, University of Massachusetts, Amherst, MA (July 1992).

[26] Swinehart, D. "The Connection Architecture for the Etherphone System", Xerox PARC, Palo Alto, CA (Dec 1990).

[27] Sun Microsystems, Inc., "Remote Procedure Call: Protocol Specification", RFC 1057, Sun Microsystems, Mountain View, CA.

[28] Topolcic, C., "Experimental Internet Stream Protocol", RFC 1190, IETF CIP Working Group (Oct 1991).

[29] Touch, J., "Multiparty Connections", Internal Memo, USC/ISI, Marina del Rey, CA (Nov 1992).

[30] Turletti, T., "H.261 Software Codec for Videoconferencing over the Internet", Research Report 1834, Institut National de Recherche en Informatique et en Automatique, Sophia-Antipolis, France (Jan 1993).

[31] Vin, H.M., Zellweger, P.T., Swinehart, D.C., Rangan, P.V., "Multimedia Conferencing in the Etherphone Environment", *IEEE Computer*, Vol.24, No.10 (Oct 1991).

[32] Weider, C., Reynolds, J., "Technical Overview of Directory Services Using the X.500 Protocol", RFC 1309 (Mar 1992).

[33] Zhang, L., Deering, S., Estrin, D., Shenker, S., Zappala, D. "RSVP: A New Resource ReSerVation Protocol", Xerox PARC, and USC, Paper in preparation.